Philosophy of AI
Vol. 2, 2026, 23-42
DOI: 10.18716/ojs/phai/2026.11721


# NEUROMORPHIC HARDWARE AND THE PROSPECT OF AI ENBRAINMENT

*Derek Shiller*

Rethink Priorities, San Francisco, CA, United States

**Abstract** This paper presents the concept of enbrainment, argues for its philosophical relevance, and explores its connection with neuromorphic computing. Enbrainment measures the degree to which an algorithm's major computational parts are carried out by dedicated hardware subsystems. Enbrainment is a philosophically fruitful notion – it is interesting at least insofar as it may contribute to a resolution of the triviality objection to computationalism, may be otherwise implicated in robust versions of functionalism, and may also influence our understanding of personal identity. Given that human brains are functionally differentiated, with different neural circuits serving different cognitive functions, human minds count as being relatively enbrained. In contrast, the algorithms performed on von Neumann computers do not. Incorporating techniques from neuromorphic computing would help promote enbrainment in future AI systems, but whether these techniques will make sense will depend on the systems' needs. We may expect to see some differentiation between the philosophical status of local, time-sensitive robotic AI systems that benefit from enbrainment and efficient, globally available AI servers that do not.

**Keywords:** functionalism; identity; neuromorphic comptuing; von Neumann architecture; artificial intelligence

## 1. Introduction

Some recent discussions of consciousness in AI systems have suggested that possession and control of a body might bear importantly on our assessments of consciousness (Butlin et al., 2023; Chalmers, 2024; Seth, 2025). In this paper, I will define a different relation between mind and body and explore its potential significance. Whereas embodiment concerns the relation between a mind and its means of worldly interaction, 'enbrainment' will focus on the relationship between a mind and its computational mechanisms.

There is reason to think that enbrainment might be a philosophically important consideration that should inform our understanding of artificial minds. Enbrainment is a central design choice in computational engineering. When providing philosophical analyses of mental concepts in light of human minds, it is tempting to adopt enbrainment as an assumption and forget how differently computer systems function. The concept appears to underlie some prominent views in consciousness and personal identity.

Current leading AI systems are not enbrained. The standard von Neumann architecture favors flexibility over efficiency. Commercial computers have fallen on the side of general-purpose reusability. Modern scientific computing and machine learning have introduced different incentives in computational hardware. The adoption of neuromorphic techniques might make enbrainment more attractive. Depending on the needs of future AI systems, we could either see a rise in enbrained systems or their continued relegation to experimental computer science.

I will start in section 1 by outlining the concept of enbrainment, considering its role in human cognition, and motivating its philosophical significance. Then in section 2, I will describe why most contemporary computer systems are not enbrained. I will explain how certain neuromorphic computers invoke elements of enbrainment and how a trend toward neuromorphism could be expected to bring with it a shift toward greater enbrainment. In section 3, I will speculate about possible pressures on AI hardware and how some push in the direction of enbrainment while others push against it. The kinds of architectures we see (and in which contexts) will depend on the needs and constraints of future AI systems.

## 2. What is enbrainment and why care?

### 2.1 What is enbrainment?

We might think of minds as implementing certain cognitive algorithms (its software) within a certain physical structure (its hardware)[1]. Enbrainment describes the relation between this mental software and the underlying hardware.

> **Enbrainment:** A system implementing a cognitive algorithm is enbrained to the degree that the algorithm's major computational parts are played by specific hardware subsystems. The same functions are subserved by the same physical components from moment to moment, no component subserves many radically different functions, and the components are dedicated to playing their role specifically to that system over its life.

This definition combines several distinct requirements.

The first requirement is that the same physical components play specific computational roles for the system over time. Certain parts of the componentry are responsible for ensuring that certain parts of the algorithm happen. They may embody algorithmic mechanisms or data structures. The same parts of the system's componentry are responsible for the same parts of the algorithm time after time.

Consider an application-specific integrated circuit designed to efficiently perform the SHA-256 hashing algorithm in Bitcoin mining. Individual parts of the hashing process are managed by specific circuits whose physical structure is fine-tuned for the computations they need to perform. Such a circuit effectively *enbrains* the SHA-256 hashing algorithm: the circuit parts are dedicated to the roles they play, maintaining consistent physical components for each algorithmic step rather than reusing general-purpose hardware for different ends.

Second, the definition requires that the components that are responsible for the relevant algorithmic parts are allocated primarily to a single computational system. The componentry that plays specific roles isn't shared between many different systems.

1. It may not be easy to separate hardware and software (Kleiner, 2024). I take this to be a conceptual possibility, even if in practice, hardware details determine or constrain possible software.

Consider the random number generation service, such as RANDOM.ORG (Haahr, 2025). This service is dedicated to a single computational function: generating truly random numbers using atmospheric noise rather than algorithmic pseudorandom number generation. The same physical hardware, such as radio receivers detecting atmospheric static, analog-to-digital converters, and processing circuits, consistently performs this specialized task. However, this hardware simultaneously serves many different systems: researchers conducting statistical analyses, game developers creating unpredictable gameplay elements, cryptographers generating secure keys, lottery systems selecting winners, and countless web applications requiring randomness. While the componentry is functionally dedicated to randomness generation, it is not dedicated to any particular cognitive or computational system. Each random number request might serve a completely different application with entirely different purposes. In this case, the hardware maintains consistent computational roles but lacks the system-dedication that full enbrainment requires.

These ideas are broadly applicable to computer systems capable of implementing algorithms. The application of the concept of enbrainment to *minds* requires several further assumptions about the nature of minds.

First, the application to minds assumes that possessing a mind involves performing characteristic computations (grouped together in algorithms). For computationalists (Piccinini, 2010), this is all that a mind needs to do. Any system that implements the right algorithms counts as having corresponding mental states. These mental states might be conscious episodes with phenomenal contents, but they might also be attitudes like belief or desire. However, the definition of enbrainment does not assume computationalism to be true. Whether or not one accepts computationalism, one can accept that minds perform computations and that certain algorithms capture the major activities of cognition – e.g., that the behavioral upshot of cognition is fixed by its associated algorithms and any other ingredients of cognition are correlated or epiphenomenal. Even non-computationalists can accept that there are relations between the computations performed by a mind and the physical structures underpinning them.

Second, the definition of enbrainment assumes that the relevant algorithms that minds compute are structured entities. This is a familiar concession to common-sense descriptions of algorithms, which typically describe a sequence of processes carried out on component elements (Hill, 2016). The component parts of algorithms may include, among other things, memory locations, data structures, and subroutines. For instance, a sorting algorithm like mergesort may describe processes on an array. It may indicate that separate arrays are created and an identical process is carried out on each of them. It may call for elements to be inserted or removed, which in application may involve further algorithms operating at a lower level. Both the operations and operanda may be considered parts of the algorithm.

We rarely explicitly specify the boundaries of data structures and subroutines in algorithms; people may disagree about the structure of specific cases (Blass et al., 2009). Computationally, it may not matter whether a certain algorithmic subroutine is played everywhere by metaphysically identical physical implementations as opposed to algorithmically identical implementations. Nevertheless, it is quite plausible that natural descriptions of the algorithms implemented by minds will make reference to certain parts or processes and that they can, for the purposes of assessing enbrainment, be regarded as having consistent identities over time. A global workspace algorithm, such as is implicated by Global Workspace Theory (Baars, 2005; Dehaene & Naccache, 2001; Mashour et al., 2020), is naturally interpreted as positing one structure, the workspace, that is used as part of a filtering and broadcasting cycle. Enbrainment with respect to the global workspace algorithm means that the same physical structure is used as the workspace over time, and the workspace serves as such for only a single mind, rather than workspaces being constructed anew or requisitioned from elsewhere during successive broadcasting cycles.

Isomorphic accounts of computation implementation (Chalmers, 1996; Putnam, 1988) rely on mappings between the abstract structure of the algorithm and some candidate for implementation. Indeed, a system whose parts were so fundamentally mixed that its processes and data structures were inextricably holistic could not count as enbrained. However, there are ways of implementing algorithms where there would be a mapping between physical processes and algorithmic parts without enbrainment. This might be because different physical structures take up the role of performing the same parts of the algorithm at different times. The very same computational entities may be composed of different physical mechanisms each moment.

As defined, enbrainment bears a strong resemblance to the 'integrity constraints' I previously proposed (Shiller, 2025). These constraints were proposed as requirements for the proper implementation of algorithms for certain philosophically robust cases, such as in assessing consciousness according to functionalism. The idea was that some amount of internal integrity was required among the parts of a system that play functional roles. I developed specific constraints on each part's persistence, their relation to the underlying material basis, and their characteristic causal powers. The concept of enbrainment aims to highlight a general quality of computational systems that would contribute toward systems having the right integrity, with enbrained systems being likely, but not guaranteed, to satisfy such constraints. Despite this general aim, enbrainment may be an interesting concept even for those who are skeptical about integrity constraints for consciousness in particular (such as (Dung & Kersten, 2025)) because of its other potential applications.

Since enbrainment depends on a set of factors that can be satisfied to varying extents, it is a property that comes in degrees. The extent to which there exist physical mechanisms for specific functions can vary from function to function and moment to moment.[2] Insofar as multiple brain mechanisms might handle the same functions at different times, or some physical circuits might be involved in different functions, human brains are not maximally enbrained.

I will use the term 'enbrained' to describe systems that have a modest level of enbrainment, such that there are at least some physical components that play a fairly narrow range of important functions somewhat consistently over time. The looseness of this term will not prevent us from drawing a clear distinction between the brains of animals and contemporary computers.

### 2.2 The case for human enbrainment

Before elaborating on why enbrainment might matter, I will present a brief case for thinking that human cognition is enbrained. The reasons to care about enbrainment will stem from its potential contribution to making sense of facets of our mental lives. If it turned out that our minds were not enbrained, we would have less reason to think that the concept held any philosophical weight.

---

2. There could also be variance in the amount of work components perform for different systems: if some brain components were used to perform a different function entirely outside of the mind while idle, that would lower enbrainment. The SETI@home program (Korpela et al., 2002) utilized idle private personal computers for data processing. We might imagine a future version of that program could make use of unused neural circuits in the human mind through some sort of collective interface. A neurological SETI@home-like program could utilize the distributed neural circuits of sleeping volunteers to contribute to a centrally-orchestrated algorithm. It is hard to get a grip on whether such a setup would allow for some sort of aggregate mind distributed among the co-opted circuits or whether the participants would have phenomenal experiences as they contribute computations. Enbrainment seems central to what we should make of this thought experiment.

Our cognitive lives revolve around the workings of our brains, and our brains are dedicated to running *our* minds. Unlike modern processors, which can handle many different programs and operating systems, our brains are limited to human cognition, and human minds cannot (yet) travel from brain to brain. The materials in our brains may shuffle in and out over time, but we rely moment to moment on a single brain composed of persistent neurons.

Enbrainment further requires some functional dedication at lower levels than the whole brain. Parts of our brains must be responsible for parts of the cognitive algorithms involved in cognition. Here, it is less immediately obvious that human minds are enbrained. There has been a longstanding debate in the history of neuroscience about the extent of specialization in the human brain (McCaffrey, 2023). The most prominent member of this school, Karl Lashley defended the view that neurons were not dedicated to specific functions, but contributed together in a way that was more or less interchangeable (Lashley, 1929). While Lashley has had a lasting impact, much of 20th-century neuroscience has proceeded on the assumption (and given support to the conclusion) that regions can be specialized. Some have even suggested this pattern extends to individual cells (Bowers, 2009). This prevailing consensus has been challenged, with recent critics casting doubt particularly on the localization of function to specific brain regions (Anderson, 2014; Noble et al., 2024; Pessoa, 2022; Westlin et al., 2023).

The doctrine of localization holds that recognizable computational activities can be traced to specific brain regions. This suggests both specialization of those regions and, perhaps, dedication to a small set of tasks. However, the concept of enbrainment is somewhat orthogonal to this debate. It is possible that cognitive functions belong to specific dedicated neural circuits[3] that are not localized but rather distributed across multiple brain areas (Mundale, 2002). It is possible that specific neurons are involved in different circuits and contribute in the same way to many distinct functions, consistently playing the same part in all of those functions (Anderson, 2014). It is possible that different regions play distinct but highly contextual roles consistently over time, so that functionality is contextually consistent even if it is quite difficult to stabilize contexts enough to see it. While the conceptual issues in this domain are complex, the controversy around localization in human brains doesn't cast clear doubt on enbrainment.

Overall, there is substantial evidence that many functions are performed by specific, specialized brain areas and that specific circuits contribute over time to similar cognitive functions.

On a fairly coarse level, it is well-established in neuroscience that broad areas of our brains are and remain specialized to particular ends. Different brain regions exhibit different patterns of neuron morphology and connectivity. Famously, Korbinian Brodmann and his successors were able to identify significant cellular differences across separate cortical brain areas through close examination of dead cells, each with distinct cytoarchitectural features like cell density, laminar organization, and neuron types (Zilles & Amunts, 2010). It is hard to imagine that these observable differences aren't shaped by distinct responsibilities.

Differences in structure are reflected in observable differences in function (Gazzaniga et al., 2019). For instance, the visual cortex is dedicated to visual processing. Within it, the V1 region handles low-level visual features such as edges and orientation with a retinotopic map. Elsewhere, the fusiform face area specializes in facial recognition. The pre-motor cor-

3. Here we might distinguish between dedication of circuits and dedication of the neurons that compose them. It is possible for groups of circuits to be collectively geared toward specific functions, even if those neurons are utilized in different ways across different circuits. My definition of the concept of enbrainment does not take a stand on which form of dedication matters.

tex is involved in action planning. The cingulate cortex contributes to evaluating emotionally and motivationally salient information. These regions perform their functions consistently across seconds, minutes, days, and even years. They display increased activity during task-related engagement, as can be seen on an fMRI or PET scan. Lesions to them often produce persistent related deficits, as we would expect if specific physical mechanisms were dedicated to specific functions. Direct stimulation can sometimes even cause related phenomena, like visual hallucinations or motor responses.

This is not to say that the evidence is completely straightforward. Many brain regions are associated with different functions that can be hard to tease apart. The cingulate cortex, for instance, is associated with a variety of functions such as emotion, vocalization, autonomic control, social relationships, decision-making, memory, and attention (Devinsky et al., 1995; Leech & Sharp, 2014; Vogt, 2016). These functions may all be associated, to varying degrees with a core theme around emotional control of behavior (pain, learning, deliberation, motivation, etc.), but the range of functions still may be taken to tell against enbrainment.

We may see the apparent functional overlap revealed by neuroscientific studies disappear as we look at finer resolutions and separate regions into more specific subregions (e.g., see (Foster et al., 2023; Kollenburg et al., 2025) for some functional subregions within the cingulate cortex). Individual neurons and neural circuits may be dedicated to relatively specific ends, even if they are housed close to others with different purposes. Detecting these distinctions would require a quite sophisticated understanding of what each neuron is doing. While we are far from the comprehensive understanding of brain functioning we would need to resolve these questions, the partial regional localization we have found so far might reasonably make us optimistic that we will someday assign specific cognitive roles to specific neural circuits.

Nevertheless, as we study the brain in greater detail, we should also not be too surprised if trends toward role non-specificity, parallel function implementation, and neural plasticity over time continue, each of which would count against enbrainment. The extent of human enbrainment might also depend on how we carve up the functions the mind performs: with some carvings, we may see common-sense cognitive functions broken down into an ontology of widely-shared submechanisms. The extent to which the human mind would then count as enbrained would depend on whether we think those submechanisms should be counted as part of its computational identity as opposed to merely the details of some implementational substrate. If the functions of those mechanisms are suitably cognitive and are performed by dedicated neural circuits, then the mind would be enbrained even if they are used for very different high-level tasks. If those mechanisms are sub-cognitive, then the reuse means that the same hardware is put toward different cognitive ends. The distinction between the cognitive and sub-cognitive could be partly a matter of taste.

The overall body of evidence falls short of clearly establishing that brains implement algorithms and circuits in the brain that correspond to parts of those algorithms. The brain is not a homogenous unit; differences in wiring and neuron morphology plausibly underlie the functional specialization we see in experiments. It isn't even completely clear that minds perform computations (Maley & Shagrir, 2025). Still, the debate over human enbrainment is surely one of degrees, and almost no one in the debate is committed to a wholesale rejection of enbrainment in which functions are radically disassociated from specific neural underpinnings.

This is weighty enough reason to consider the potential significance of enbrainment and think about the ways artificial systems might differ.

### 2.3 How could enbrainment matter?

Enbrainment is worth exploring not only because it represents a fundamental dimension along which computer architectures may differ, but also for the potential bearing it has on key philosophical questions. I will here suggest that we might believe that the physical makeup of brain states plays a role in setting the identity conditions of mental states, and this might complicate the attribution of mental states to non-enbrained systems. My approach will be to identify existing topics that raise issues clearly relevant to enbrainment, even if the authors did not conceive of them in that light.

I will propose three ways that enbrainment might matter to mental-state identity.

First, something in the vicinity of an enbrainment constraint on mentality is implicated in some attractive resolutions to the triviality objection to computationalist theories of mind.

Computationalism suggests that we should think about mental states in terms of algorithms. To be capable of a given mental state is to be a system that computes a certain algorithm. To occupy that mental state is to inhabit some position in the processing of that algorithm: perhaps with specific settings to variables. For instance, to be conscious of a red square might be to have a representation that encodes redness and squareness bound together and is broadcast from one's global workspace (Carruthers, 2019).

Computationalists have historically accepted fairly liberal approaches to understanding the implementation of algorithms (Shiller, 2025) such that any system with the right structure of state-transition counts. The triviality objection (Piccinini, 2010; Putnam, 1988; Sprevak, 2018) holds that there is no good way to distinguish the complex systems that do and do not implement any given algorithm because it is too easy for the states of a complex system to be mapped to corresponding states of an algorithm. The problem arises from the assumption that arbitrary ways of defining states in the system are taken to be sufficient, and there are many ways to group states of a physical system to make that system look like it is implementing different algorithms (Chalmers, 1996).

Triviality poses a problem for the philosophy of computation in general, but it holds special significance for computationalism about consciousness. We may be able to avoid triviality by holding computations to be partly conventional, such that we treat certain physical devices as computers because that is what we use them for, or made them for. It is less acceptable to apply this solution to phenomenal experiences by treating them as a matter of convention. If the objection proves sound, then implementing an algorithm turns out to be too easy to do important explanatory work — any sufficiently complex physical system could be interpreted as implementing any computation. Searle (Searle, 1992) argued that even the molecules in a wall could be interpreted as implementing the Wordstar word processing program, while Putnam (Putnam, 1988) showed how to construct mappings that would make a simple physical system appear to implement any finite state automaton. Critics have argued that this renders computational theories of mind vacuous, since they cannot meaningfully distinguish systems with cognitive states from those without.

Introducing a requirement that conscious states be enbrained may help to resolve this problem, or may be implicated in other resolutions. For instance, in his influential discussion of the problem, David Chalmers ((Chalmers, 1996), p. 330) suggests the following:

> In order to have a true implementation of a CSA [combinatorial state automaton, e.g., multi-part computational system], we must require that the various parameters are in some strong sense *independent*, corresponding to separable components of a system. The

> easiest way to do this is to require that for a system to implement a CSA, the physical differences relevant to the variation in a given parameter must be restricted to a limited physical region, with different physical regions for each element. That is, the values of a "parameter" supervene on a distinct region for each parameter.

Chalmers withholds full endorsement of this approach, noting only that this is the *easiest* way to secure independence. He is hesitant to rule out systems we might otherwise think of performing computations on the basis of this constraint. I take it that his solution is uninteresting unless there is a kernel of plausibility to the restriction and that it warrants being taken seriously as a starting point for a more complete resolution. Having separate physical components that are dedicated to representing the different parts of the algorithm contributes to the independence offered by separate physical regions. It may be that we can solve the problem with other constraints, but Chalmers' proposal is natural and plausible, and if we opt for other constraints, we should weigh them against Chalmers' proposal.

Another strain of response to the triviality objection holds that computations must be implemented in a natural way (Godfrey-Smith, 2009). If we take this alternative seriously, we might expect that the contours of the physical system should match the contours of the cognitive algorithm it implements. In other words, we might want to make sure that the physical substrate isn't gerrymandered into satisfying the relevant properties.

One way to ensure such contours match is to require some alignment of the boundaries between physical and algorithmic parthood. In order for a system to count as implementing a part of an algorithm, it must have a natural physical part whose behavior corresponds to the affordances of that part of the algorithm. The physical mechanisms should carve up the system in ways that mirror how the algorithm divides its processes into subtasks. If there is a variable used by the algorithm, we might expect there to be a consistent physical signature of the variable in the system. If there is a subroutine, we might expect it to invoke the same physical mechanisms from application to application.

There are surely other ways to naturally implement an algorithm, but enbrained algorithms are strong candidates for this kind of natural implementation. While naturalness comes in degrees, which may make us hesitant to give it much work in determining a seemingly binary property like computation implementation (Chalmers, 1996), it provides useful guidance for distinguishing genuine computational systems from mere formal mappings. The requirement that physical and computational structure align helps rule out many of the trivial implementations that motivated the objection in the first place.

Second, enbrainment may be a requirement of certain metaphysically robust versions of functionalism.

Functionalism is a broader theory than computationalism (i.e., computational functionalism), although the two are seldom clearly distinguished (Piccinini, 2010). Functionalists accept that the mind is a product of a system's functional organization. Computationalists think that what matters is that the system implements a certain algorithm, regardless of how that is achieved. One way these can come apart is if the relevant aspects of organization don't relate to the implementation of specific algorithms. Another way these can come apart is if the organization is taken to apply to the system's physical structure, rather than its abstract computational structure.

Some functionalists might place greater weight on the physical boundaries within a system, allowing that it counts as having a specific functional organization only if the system's recognizable parts align in the right ways. By requiring assignment of function to parts, enbrainment encourages that sort of structural identity.

Consider Lewis's ((Lewis, 1972), p. 256) Ramsey-inspired understanding of psychological functionalism:

> Think of common-sense psychology as a term-introducing scientific theory... Collect all the platitudes you can think of regarding the causal relations of mental states, sensory stimuli, and motor responses…
>
> Form the conjunction of these platitudes; or better, form a cluster of them – a disjunction of all conjunctions of most of them. (That way it will not matter if a few are wrong.) This is the postulate of our term-introducing theory…
>
> From the postulate, form the definition of the T-terms; it defines the mental states by reference to their causal relations to stimuli, responses, and each other. When we learn what sort of states occupy those causal roles definitive of the mental states, we will learn what states the mental states are.

This theory says that we can identify mental states by formalizing our theory into a description of relations between unlabelled entities and looking at what entities exhibit the right inter-relations. To apply the Ramsey-sentence approach, we need to compare our formalized theory against an existing ontology populated by entities and their relations. We can look to those entities to see 'what sort of states occupy those causal roles.'

As we saw with the triviality objection, a liberal ontology might lead to an explosion of satisfiers. Even if we're not specifically concerned about letting too many things count as having minds, we might think that it goes against the spirit of the view to allow the relevant states to be arbitrarily gerrymandered. In particular, we might want potential role-realizers to have their own identity and for that identity to be independent of (and therefore theoretically pre-exist) the role that they play within the cognitive system.

There are several ways of interpreting functionalism, corresponding to what I (Shiller, 2025) have called 'pattern-first' and 'system-first' functionalism. On the system-first approach, functionalism is tasked with distinguishing those systems that have minds from those that don't. We assume that the systems have a pre-existing coherence that makes them a proper candidate for having a mind. Mind-like relations between disparate objects can't be enough to make them into a candidate for having a mind. We may extend this perspective to propose that the potential realizers of functional roles within each system also need to have a proper role-independent identity. Functionalism says that such systems have minds only if their parts exhibit a certain kind of relationship with each other, and that the parts have to exist as coherent entities independent of such relations.

In contrast, on a patterns-first approach, functionalism is tasked with explaining what patterns to look for in the world in order to locate minds rather than which systems might count as having minds. The patterns that underlie minds can take place between complex and diverse sets of objects, and it can be only in terms of their relationships that they count as constituting a single coherent system.

The present claim to the philosophical relevance of enbrainment draws inspiration from these two ways of understanding functionalism. It depends on a related ontological question, which concerns the degree to which the relevant parts of a system might be abstractions over the concrete materials that make it up. The logic of digital entities within a computer system is grounded in the machine code and the physical structure of the processor, but the grounding relation is quite complex, and it is better to think of variables and subroutines as having convoluted and inter-defined identity conditions. In the same way that we may ask whether arbitrary groupings of objects might count as mind candidates on account of their inter-relations, we may ask whether abstractions over computer logic can count as role-player candidates.

In an enbrained system, function dedication suggests that there are material objects that can count as the realizers of functional roles. In an un-enbrained system, the material realizers of roles may change frequently over time. Their persistent identity as a part of the system capable of sustaining ongoing relationships depends in part on the same structures that make them realizers of those roles. Patterns-first functionalists may be more amenable to finding the relevant patterns among objects in an ontology that is largely abstracted from the implementational details. Systems-first approaches may lean toward being more discerning in the implementational details, if they expect the identity conditions of the functional relata to also be non-relational.

Enbrainment doesn't guarantee that the component parts of a brain that are dedicated to specific parts of an algorithm will count as entities in our ontology; non-enbrainment doesn't guarantee that abstractions over changing material componentry won't. Still, it is reasonable to expect that dedicated computational componentry will offer up better candidates for independently real role realizers. In a non-enbrained system in which the parts are subserved by common computational mechanisms, we should expect the relevant computational entities that play functional roles to require fairly elaborate definitions in terms of the underlying machinery, given that the machinery can be used to play so many different parts (including possibly parts of different minds). In a standard von Neumann architecture, the status of the algorithmic components that make the system count as being in different states depends upon lengthy pointer chains and the proper interpretation of many memory cells. They do not seem like the kinds of things that have an existence independent of the computations that they support.

Computationalism is typically interpreted in line with patterns-first approach, though there are plausibly some readings of it that adopt the perspective of the systems-first approach. We could identify consciousness only with coherent systems of good ontological standing implementing the right algorithms in a straightforward way. Given the popularity of computationalism among functionalists, we should perhaps not expect to see much interest in systems-first approaches or much concern about enbrainment. However, the popularity of computationalism may result from the fact that issues like these have been insufficiently discussed; focusing more on the differences between enbrained and unenbrained computer systems might help bring them to light.

Third, some theories in the philosophy of personal identity place importance on the continuity of the physical components that make up a person.

One version of this view, defended first by Peter Unger (Unger, 1990), says that what matters to our survival is the continuity of the materials underlying our core psychology: those psychological capacities that are central to being a cognizing thing. Core psychology is not what distinguishes us from each other, so for Unger, it doesn't matter to our survival if our memories are erased or our personality changes. What matters is that we maintain the physical basis for basic mental capacities like awareness, reasoning, and experiencing. According to this view, persistence requires that the physical substrates that enable cognition survive, even if their specific contents change.

Jeff McMahan (McMahan, 2002), Dainton and Bayne (Dainton & Bayne, 2005), and Duncan (Duncan, 2020) defend related views focused on the persistence of a specific faculty: the faculty underlying consciousness. If the material basis for our consciousness persists, then we will survive.

Dainton and Bayne (Dainton & Bayne, 2005), p. 567) summarize their view:

> A person persists through a period of unconsciousness by virtue of retaining the capacity for phenomenal consciousness; provided a person's capacities for consciousness endure, so too does the person. Now of course, generally speaking, capacities are possessed by objects, and it is this assumption that leads to the view that persons are phenomenal substances in the sense just specified: the latter being nothing other than objects which possess the capacity for generating phenomenally continuous consciousness.

These kinds of views make less sense if there is no physical componentry that is dedicated to performing specific roles within our mental lives. Suppose that we took Lashley's equipotentiality doctrine to an extreme and imagine that each neuron was functionally identical to each other, such that any given mental faculty could be replicated easily in any sufficiently large lump of neural flesh. The neural matter that implements our core psychological capacities (or consciousness in particular) at one moment might not at the next. Brain matter that sits completely inactive at one moment might take on full responsibility for all of the brain's machinations the next. The capacities would not be tied to specific circuits and could dance across the brain. The physical continuity of any single, formerly inactive part of the brain would seem to be irrelevant, and so we should wonder what kind of physical continuity was required at all or how to understand the proposals when responsibilities so easily shift.[4]

If human minds are enbrained, then we might expect to preclude persistent identities from any systems that are not relatively enbrained. The mechanisms underlying core psychology and that support consciousness in such systems may differ each moment. Insofar as persistence of those faculties is what matters, different computational substrates at different times will make for different individuals.

Of course, it could be that the right lesson from these applications is that we shouldn't take such views about personal identity seriously, that we might need to flesh out an ontology in which computational abstractions are proper entities (rather than just hardware parts) with substantial identities independent of the algorithms in which they figure, and that we need a better resolution to the problem of triviality that doesn't depend on enbrainment. I take it that these kinds of conceptual connections, originally advanced with little attention to von Neumann architectures, neuromorphic computing, or the concept of enbrainment, make it sufficiently interesting to be worth deeper dedicated exploration. Even if we're yet to be convinced about the substance of any application, they suggest that enbrainment may carry significant philosophical weight, and we should pay attention to it when we weigh the differences between human and AI cognition.

## 3. Enbrainment in current and neuromorphic computer hardware

### 3.1 Von Neumann systems

Most contemporary general computer systems follow a von Neumann architecture (Hennessy & Patterson, 2012) in which program instructions are stored as data in general-purpose memory and are run on general-purpose processors. Computers rely largely on CPUs for managing sequential operations within computer programs, and they use the same handful

4. Dainton and Bayne allow that it is possible that capacities might survive the changing of their material basis. It is conceivable that one could have a view on which computational capacities might move between systems, being the same capacity rather than a copy. This would drive a wedge between the view that identifies persons with capacities for consciousness and persons with objects possessing those capacities. It seems far simpler to identify persons with the objects.

of CPU registers for nearly everything, from maintaining process permissions, memory allocation, and thread priority within an operating system, to handling event interrupts, issuing commands or delegating work to external attachments, and performing the steps for all manner of programs run on the system. Even specialized GPUs perform relatively generic instructions and are indifferent to precisely what functions they are computing.

Computer programs leverage memory-processor interactions to coordinate operations, including manipulations of data structures and invocations of methods. The life of these objects within the computer hardware is ephemeral and largely abstract (Aho et al., 2007). The process of translation of the computer program through compilation and translation to machine language ensures that the system acts as if it were manipulating a set of persistent data structure objects, but these objects have no fixed physical manifestation. Boundaries aren't marked; rather, each program is translated into a form in which all the memory accesses are correctly configured to look for relevant data at the right memory locations. Variables, arrays, and other data structures exist primarily as abstractions mapped onto arbitrary materially identical transistor circuits.

Memory consists of a set of completely generic stores organized in a hierarchical system. The boundaries between memory locations storing data structures are not physically marked. Memory is simply a continuous array of addressed bits. Programs are written to maintain virtual boundaries through memory management systems, standard data structure sizes, pointer arithmetic, and address translation. The physical location of memory can be moved through virtual memory paging, memory mapping, and cache operations, so long as the references (pointers, page tables, or cache tags) tracking the physical or virtual addresses are properly updated. Data may be stored across multiple levels of the memory hierarchy, from registers to various levels of cache to main RAM to disk storage, with hardware and software managing the movement of data between these levels to optimize access patterns and performance. This fluid nature of memory allocation and management means that the same algorithmic data structure may occupy different physical locations at different times during program execution.

The same processors, such as the registers in a traditional CPU or a GPU chip, may be used for many different purposes, including handling the computations for different nodes in an artificial neural network or running the same AI model on different inputs. GPUs contain many processor cores that can execute thousands of parallel threads, dynamically allocating computational resources to different tasks. During neural network inference, the same physical cores may process matrix multiplications for different network layers, activation functions, or batch members in rapid succession. The hardware remains functionally generic, with the same physical processing units being repurposed for whatever computation is needed, whether that's convolution operations, attention mechanisms, or linear transformations.

Contemporary AI models are often housed on data center servers and accessed through repeated API requests, allowing for radically distributed processing. There is no guarantee that successive responses, or even different parts of a single response, will be run on the same processors. For large language models, it is common to break up stages into processing of given text and response generation, which have somewhat different needs and may make sense to carry out separately (Yu et al., 2022). It may also make sense to carry out different parts of the response on different models, using simpler models for some parts and more sophisticated models for others.

Because of the generic capabilities and memory flexibility, we can't find much continuity between the parts of the physical system that correspond to the hardware and the parts of the algorithm. The physical implementation in modern computers is highly abstracted and fluid. The same processors handle different algorithmic functions from moment to moment, data structures exist only virtually and can be relocated across memory, and there is no stable mapping between computational elements and specific hardware components.

This stands in stark contrast to biological brains, where dedicated neural circuits may consistently handle specific cognitive functions over time. The ephemeral nature of computational processes in modern von Neumann architectures means these systems lack the kind of physical dedication and continuity that characterizes enbrainment.

### 3.2 Neuromorphic Systems

Neuromorphic techniques in computer hardware have drawn inspiration from the ways that animal brains function, ranging from basic computational component functions to larger aspects of architecture (Tsur, 2021). While mainstream AI neural networks are loosely inspired by the brain, they have largely abstracted from many details of neural behavior that don't fit well with traditional hardware. Neuromorphic techniques can enable different abstractions, such as making the temporal complexity in spiking neural networks computationally tractable.

Adopting some of the characteristics of animal brains will bring others along. While engineers may not be focused on enbrainment as an end goal, many of the characteristic features of human brains are built around enbrainment and make sense in that context. Even if neuromorphic computers are not designed with enbrainment in mind, we can expect that importing other features of human brains will lead us to produce cognitive systems with greater enbrainment.

Several key neuromorphic techniques contribute to increased enbrainment. Should they become more common, we can expect future computer systems to be more enbrained.

First, state-of-the-art neuromorphic systems typically provide large numbers of processors with their own persistent local memory. This helps to avoid the "von Neumann bottleneck" (Backus, 1978). CPUs contain a relatively small number of registers organized into cores. Each core manages a custom stream of instructions that is fetched from RAM and stored temporarily in a cache. A multi-core processor can perform the computations for a small number of different programs simultaneously, with each core acting independently and complex mechanisms to keep caches in sync. GPUs are specialized to run the same instructions in parallel on a number of different datapoints, allowing a different sort of parallelizability. GPU programs must carefully set up data structures so that the computations can be performed in a rigidly structured way. Both sorts of processors reflect the limitations of von Neumann architectures. It is challenging to give many different processor units dynamic access to data, so it makes sense to either limit the number of processors or rigidify their data access.

Sophisticated neuromorphic chips like IBM's TrueNorth (Akopyan et al., 2015), Intel's Loihi (Davies et al., 2018), and the University of Manchester's SpiNNaker (Furber et al., 2013) introduce processors that are configured to work from settings saved in a local memory. They operate not by fetching and carrying out instructions from RAM, but through exchanging simple messages and processing them in light of their locally stored setup. This localization of data or instructions allows for more customized variation of parallel processor behavior without requiring an unworkably complex data pipeline. Given that nodes have their own memory and can only communicate with brief messages, they must largely maintain their roles over time. Enbrainment is a result of minimizing data transfer.

Second, the neuromorphic architectures that localize data (such as Loihi, TrueNorth, and SpiNNaker) often implement sophisticated communication pathways between processing elements that fundamentally differ from conventional von Neumann architectures. Rather than processors communicating through modifying the data stored at shared memory addresses in RAM, these systems enable message-passing between processing units, mimicking the spike-based communication of biological neurons.

These many-to-many communication networks require complex data transmission infrastructures to orchestrate the right connections at the right times. The implementation typically involves one of two approaches: dedicated physical communication channels between specific processing elements, or programmable routing hardware configured with connection patterns during system initialization. In both cases, the resulting communication complexity naturally favors the emergence of specialized, persistent functional relationships between processor nodes.

The direct communication pathways offer significant advantages over traditional centralized bus architectures by enabling parallel operation, which reduces latency and removes bottlenecks. These connections can be implemented as fixed routing networks between processing elements, creating a mesh-like connectivity pattern. This design philosophy enables local processing clusters and communication patterns that closely mirror biological neural circuits—nearby processing units maintain dense interconnections for rapid local computation, while selective longer-range pathways facilitate information integration across different functional regions.

The physical dedication and persistence of these communication channels contributes directly to enbrainment by establishing stable structural relationships between components. These hardwired or pre-configured connections create consistent information flow patterns that shape how the system processes and integrates information.

Third, some research has focused on blurring the line between memory and computation. Optical neural networks can encode network weights in hardware that directly modulates network activations passed through as light (Xu et al., 2021). Memristors (Zidan et al., 2018), on the other hand, make it possible to build circuits whose behavior is dependent on their past activity. Unlike traditional computing architectures, where memory storage and processing occur in separate components requiring constant data transfer, these techniques achieve computational-memory integration at the hardware level.

Implementing combined memory and logic functionality in hardware can help with running neural networks (Matuszewski et al., 2024; Thomas, 2013), particularly in reducing the hardware overhead required to manage network state. It also leads the networks to be enbrained, as storing memory in processor circuitry ties that circuitry to those functions that need the data that are in that memory.

These three hardware architectural features can provide significant efficiency advantages. By maintaining dedicated hardware for specific functions, systems can avoid the energy and time costs of constantly moving data and instructions between generic components. Direct communication pathways minimize routing complexity and bottlenecks. The physical persistence of computational elements reduces the overhead of repeatedly setting up and tearing down abstract logic.

While current leading neuromorphic systems are still experimental and are designed for exploring capabilities rather than building commercial products, their characteristic features represent important steps in the direction of enbrainment. The efficiency benefits of these approaches suggest that increasing enbrainment may be valuable for future AI systems, even if consciousness or identity are not the primary goals. The physical dedication of persistent, pre-configured memory in processors reduces energy costs from constantly moving data between components. Direct communication pathways minimize latency and routing overhead and support greater parallel computations. Persistent physical states through specialized memory-processor circuits allow for more efficient information storage.

## 4. Future Directions for Enbrainment

The disparity between enbrained biological cognition and non-enbrained artificial intelligence is not merely a historical accident. There are substantial economic and technical reasons why current AI systems have developed within non-enbrained hardware architectures, and these factors will continue to shape the future landscape of artificial cognition.

The hardware best suited for running contemporary AI systems has grown out of a computational paradigm that prizes flexibility. This focus on flexibility has proven extraordinarily valuable for AI. The same GPU clusters that train transformer-based language models can be rapidly reconfigured to handle computer vision tasks, reinforcement learning algorithms, or other scientific computing workloads. Data center resources can be shifted to support new products or meet changes in customer demand for existing ones. GPUs can be mass-produced and sold to a variety of consumers with a variety of needs. This adaptability has been crucial during a period of rapid algorithmic innovation, where new architectures like transformers, diffusion models, and mixture-of-experts systems have emerged in quick succession.

Neuromorphic hardware, by contrast, is fundamentally tied to systems with specific architectures. This specialization means that neuromorphic hardware cannot be easily reallocated to different forms of neural networks, let alone alternative uses in scientific computing, making substantial investments in building sophisticated neuromorphic hardware more risky. This risk is amplified by the current pace of AI research. The field has seen dramatic shifts in dominant architectures over relatively short time periods. Neuromorphic hardware designed around today's assumptions about optimal neural architectures might be poorly suited to future developments.

Even if we do arrive at a future where hardware is optimized for running specific models, there are fundamental downsides to neuromorphic setups that make them less efficient in aggregate deployment scenarios. Neuromorphic architectures are often not designed to handle multiple intermingled network computations the way that GPUs can efficiently handle batches of neural network computations.

Modern GPU architectures excel at parallel processing of multiple inference requests or training batches. A single GPU can simultaneously process dozens or hundreds of different inputs, often stored as high-dimensional tensors, through the same neural network. This batch processing capability is crucial for efficient utilization of expensive hardware. The same physical processors that handle one user's query can immediately switch to processing another user's request with minimal overhead.

Neuromorphic systems — particularly those that store temporary data locally — resist this kind of optimization. When computational elements are dedicated to specific functions over time within a particular network architecture, they cannot easily be reassigned to handle different applications of the same network. The persistent states that help make neuromorphic systems efficient for deployment become a liability when trying to serve multiple users.

Brains are built to serve the needs of a single creature. Servers are not. This creates a fundamental tension between enbrainment and utilization efficiency. Expensive hardware is best utilized continuously, ideally at near-maximum capacity. If enbrained neural networks are less efficient when used to handle centralized AI models serving many clients, then we might expect to see the most heavily utilized AI systems that must justify their hardware costs through constant use dedicated to running non-enbrained AIs.

The economics of cloud-scale AI deployment strongly favor systems that can maximize throughput per dollar of hardware investment. A data center serving millions of users cannot afford to have specialized processors sitting idle while demand fluctuates across different types of requests. The flexibility to reallocate computational resources dynamically across different models and users provides crucial efficiency advantages that specialized neuromorphic hardware struggles to match.

On the other hand, there are fewer downsides to using neuromorphic setups for local, dedicated models of the sort we might use for on-the-ground control of robotic bodies. If AI systems are placed inside robots, autonomous vehicles, or other embodied agents, the economic calculations shift dramatically in favor of specialized, enbrained architectures.

Local deployment often involves strict constraints on power consumption, physical size, and real-time responsiveness that favor neuromorphic approaches. Traditional von Neumann setups excel at flexible computation but consume significantly more power per operation than neuromorphic alternatives (Tsur, 2021). For battery-powered systems or applications requiring extended autonomous operation, the energy efficiency advantages of neuromorphic hardware can outweigh the flexibility costs.

Real-time constraints further favor enbrained architectures. A robotic system navigating through a chaotic environment cannot afford the latency of waiting for cloud-based processing. Dedicated neuromorphic circuits that maintain a persistent state and can respond to sensory inputs within microseconds provide crucial performance advantages for time-critical applications.

The risk profile also differs substantially for local deployments. While a data center must hedge against unknown future algorithmic developments, a robotics company can optimize hardware for the specific tasks its products will perform.

This suggests that we should expect market segmentation rather than uniform adoption of either enbrained or non-enbrained approaches. Cloud-based AI services optimizing for flexibility and utilization efficiency will likely continue relying on general-purpose architectures, while edge applications with dedicated functions and strict resource constraints may increasingly adopt enbrained neuromorphic solutions.

The philosophical implications of this segmentation may be significant. If enbrained AI systems emerge primarily in embodied, task-specific applications while general intelligence capabilities remain concentrated in non-enbrained cloud systems, this could create interesting distinctions in how we assess consciousness and personal identity across types of AI systems. The most human-like artificial agents, those with persistent bodies under local control, might also be the most likely to possess enbrained cognitive architectures.

Several technological developments could alter these economic and technical constraints over time. Advances in neuromorphic manufacturing could reduce costs and increase flexibility, potentially expanding the range of applications where enbrained systems become economically viable. New architectural approaches might bridge the gap between specialization and flexibility, creating neuromorphic systems capable of handling broader ranges of computational tasks.

The maturation of AI algorithms could also shift the balance. As the pace of fundamental architectural innovation slows and dominant approaches stabilize, the risk of investing in specialized hardware decreases. If transformer-based architectures or their successors prove durably superior for most applications, highly specialized hardware optimized for those specific architectures becomes a more attractive investment.

## 5. Conclusion

The concept of enbrainment suggests a fundamental architectural divide emerging in artificial intelligence between experimental systems with human-like levels of hardware dedication and von-Neumann-style contemporary AI systems. While economic and technical forces will likely maintain this divide, with cloud-based AI services favoring general-purpose architectures for cost and adaptability reasons, specialized applications in robotics and autonomous systems create niches where enbrained neuromorphic architectures may flourish due to their power efficiency and real-time advantages.

These divergent paths carry the potential for significant philosophical implications for consciousness, personal identity, and moral status. The most human-like artificial agents with persistent physical forms may also possess the cognitive architectures most similar to biological minds, while the most capable general AI systems remain fundamentally alien in their basic hardware organization. The choice between flexibility and dedication in computational architecture thus becomes simultaneously a choice about the fundamental nature of artificial cognition. As we approach potentially conscious AI systems, understanding these architectural implications is important to keep in mind. We should remain cognizant of the nuances between hardware choices and continue to think through the potential philosophical substance in the fine details.

## Bibliography

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (Eds.). (2007). *Compilers: Principles, techniques, & tools* (Second Edition). Pearson Addison-Wesley.

Akopyan, F., Sawada, J., Cassidy, A., Alvarez-Icaza, R., Arthur, J., Merolla, P., Imam, N., Nakamura, Y., Datta, P., Nam, G.-J., Taba, B., Beakes, M., Brezzo, B., Kuang, J. B., Manohar, R., Risk, W. P., Jackson, B., & Modha, D. S. (2015). TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *34*(10), 1537–1557. https://doi.org/10.1109/TCAD.2015.2474396

Anderson, M. L. (2014). *After phrenology: Neural reuse and the interactive brain*. MIT Press.

Baars, B. (2005). Global workspace theory of consciousness: Toward a cognitive neuroscience of human experience. In *Progress in Brain Research* (Vol. 150, pp. 45–53). Elsevier. https://doi.org/10.1016/S0079-6123(05)50004-9

Backus, J. (1978). Can programming be liberated from the von Neumann style? a functional style and its algebra of programs. *Communications of the ACM*, *21*(8), 613–641. https://doi.org/10.1145/359576.359579

Blass, A., Dershowitz, N., & Gurevich, Y. (2009). When are two algorithms the same? *Bulletin of Symbolic Logic*, *15*(2), 145–168. https://doi.org/10.2178/bsl/1243948484

Bowers, J. S. (2009). On the biological plausibility of grandmother cells: Implications for neural network theories in psychology and neuroscience. *Psychological Review*, *116*(1), 220–251. https://doi.org/10.1037/a0014462

Butlin, P., Long, R., Elmoznino, E., Bengio, Y., Birch, J., Constant, A., Deane, G., Fleming, S. M., Frith, C., Ji, X., Kanai, R., Klein, C., Lindsay, G., Michel, M., Mudrik, L., Peters, M. A. K., Schwitzgebel, E., Simon, J., & VanRullen, R. (2023). *Consciousness in artificial intelligence: Insights from the science of consciousness* (arXiv:2308.08708). arXiv. https://doi.org/10.48550/arXiv.2308.08708

Carruthers, P. (2019). *Human and animal minds: The consciousness questions laid to rest*. Oxford University Press.
Chalmers, D. J. (1996). Does a rock implement every finite-state automaton? *Synthese*, *108*(3), 309–333. https://doi.org/10.1007/BF00413692
Chalmers, D. J. (2024). *Could a large language model be conscious?* (arXiv:2303.07103). arXiv. https://doi.org/10.48550/arXiv.2303.07103
Dainton, B., & Bayne, T. (2005). Consciousness as a guide to personal persistence. *Australasian Journal of Philosophy*, *83*(4), 549–571. https://doi.org/10.1080/00048400500338856
Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., Dimou, G., Joshi, P., Imam, N., Jain, S., Liao, Y., Lin, C.-K., Lines, A., Liu, R., Mathaikutty, D., McCoy, S., Paul, A., Tse, J., Venkataramanan, G., … Wang, H. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, *38*(1), 82–99. https://doi.org/10.1109/MM.2018.112130359
Dehaene, S., & Naccache, L. (2001). Towards a cognitive neuroscience of consciousness: Basic evidence and a workspace framework. *Cognition, The Cognitive Neuroscience of Consciousness*, 79(1), 1–37. https://doi.org/10.1016/S0010-0277(00)00123-2
Devinsky, O., Morrell, M. J., & Vogt, B. A. (1995). Contributions of anterior cingulate cortex to behaviour. *Brain*, *118*(1), 279–306. https://doi.org/10.1093/brain/118.1.279
Duncan, M. (2020). A new argument for the phenomenal approach to personal persistence. *Philosophical Studies*, *177*(7), 2031–2049.
Dung, L., & Kersten, L. (2025). Implementing artificial consciousness. *Mind & Language*, *40*(3), 285–305. https://doi.org/10.1111/mila.12532
Foster, B. L., Koslov, S. R., Aponik-Gremillion, L., Monko, M. E., Hayden, B. Y., & Heilbronner, S. R. (2023). A tripartite view of the posterior cingulate cortex. *Nature Reviews Neuroscience*, *24*(3), 173–189. https://doi.org/10.1038/s41583-022-00661-x
Furber, S. B., Lester, D. R., Plana, L. A., Garside, J. D., Painkras, E., Temple, S., & Brown, A. D. (2013). Overview of the SpiNNaker system architecture. *IEEE Transactions on Computers*, *62*(12), 2454–2467. https://doi.org/10.1109/TC.2012.142
Gazzaniga, M. S., Ivry, R., & Mangun, G. R. (2019). *Cognitive neuroscience: The biology of the mind* (Fifth Edition). W.W. Norton & Company.
Godfrey-Smith, P. (2009). Triviality arguments against functionalism. *Philosophical Studies*, *145*(2), 273–295. https://doi.org/10.1007/s11098-008-9231-3
Haahr, M. (2025, December 15). *History of RANDOM.ORG.* RANDOM.ORG. https://random.org/history
Hennessy, J. L., & Patterson, D. A. (2012). *Computer architecture: A quantitative approach* (Fifth edition). Elsevier, Morgan Kaufmann.
Hill, R. K. (2016). What an algorithm is. *Philosophy & Technology*, *29*(1), 35–59. https://doi.org/10.1007/s13347-014-0184-5
Kleiner, J. (2024). *Consciousness qua mortal computation* (arXiv:2403.03925). arXiv. https://doi.org/10.48550/arXiv.2403.03925
Kollenburg, L., Arnts, H., Green, A., Strauss, I., Vinke, S., & Kurt, E. (2025). The cingulum: Anatomy, connectivity and what goes beyond. *Brain Communications*, *7*(1), fcaf048. https://doi.org/10.1093/braincomms/fcaf048
Korpela, E., Anderson, D., Cobb, J., Lebofsky, M., & Werthimer, D. (2002). SETI@home: An experiment in public-resource computing. *Communications of the ACM*, *45*(11), 56–61.
Lashley, K. S. (1929). *Brain mechanisms and intelligence: A quantitative study of injuries to the brain* (pp. xi, 186). University of Chicago Press. https://doi.org/10.1037/10017-000

Leech, R., & Sharp, D. J. (2014). The role of the posterior cingulate cortex in cognition and disease. *Brain: A Journal of Neurology*, *137*(Pt 1), 12–32. https://doi.org/10.1093/brain/awt162

Lewis, D. (1972). Psychophysical and theoretical identifications. *Australasian Journal of Philosophy*, *50*(3), 249–258. https://doi.org/10.1080/00048407212341301

Maley, C. J., & Shagrir, O. (2025). The mind-brain is a computer, but what is (neural) computation? In *Neurocognitive Foundations of Mind* (pp. 144–163). Taylor and Francis. https://doi.org/10.4324/9781003458531-8

Mashour, G. A., Roelfsema, P., Changeux, J.-P., & Dehaene, S. (2020). Conscious processing and the global neuronal workspace hypothesis. *Neuron*, *105*(5), 776–798. https://doi.org/10.1016/j.neuron.2020.01.026

Matuszewski, M., Prystupiuk, A., & Opala, A. (2024). Role of all-optical neural networks. *Physical Review Applied*, *21*(1), 014028. https://doi.org/10.1103/PhysRevApplied.21.014028

McCaffrey, J. B. (2023). Evolving concepts of functional localization. *Philosophy Compass*, *18*(5), e12914. https://doi.org/10.1111/phc3.12914

McMahan, J. (2002). *The ethics of killing: Problems at the margins of life*. Oxford University Press.

Mundale, J. (2002). Concepts of localization: Balkanization in the brain. *Brain and Mind*, *3*(3), 313–330. https://doi.org/10.1023/A:1022912227833

Noble, S., Curtiss, J., Pessoa, L., & Scheinost, D. (2024). The tip of the iceberg: A call to embrace anti-localizationism in human neuroscience research. *Imaging Neuroscience*, 2, imag–2–00138. https://doi.org/10.1162/imag_a_00138

Pessoa, L. (2022). *The entangled brain: How perception, cognition, and emotion are woven together*. The MIT Press. https://doi.org/10.7551/mitpress/14636.001.0001

Piccinini, G. (2010). The mind as neural software? Understanding functionalism, computationalism, and computational functionalism. *Philosophy and Phenomenological Research*, *81*(2), 269–311. https://doi.org/10.1111/j.1933-1592.2010.00356.x

Putnam, H. (1988). *Representation and reality*. MIT Press.

Searle, J. R. (1992). *The rediscovery of the mind*. MIT Press. https://direct.mit.edu/books/monograph/4733/The-Rediscovery-of-the-Mind

Seth, A. K. (2025). Conscious artificial intelligence and biological naturalism. *Behavioral and Brain Sciences*, 1–42. https://doi.org/10.1017/S0140525X25000032

Shiller, D. (2025). How many digital minds can dance on the streaming multiprocessors of a GPU cluster? *Synthese*, *206*(25), 218. https://doi.org/10.1007/s11229-025-05310-1

Sprevak, M. (2018). Triviality arguments about computational implementation. In *The Routledge Handbook of the Computational Mind*. Routledge.

Thomas, A. (2013). Memristor-based neural networks. *Journal of Physics D: Applied Physics*, *46*(9), 093001. https://doi.org/10.1088/0022-3727/46/9/093001

Tsur, E. E. (2021). *Neuromorphic engineering: The scientist's, algorithms designer's and computer architect's perspectives on brain-inspired computing*. CRC Press.

Unger, P. (1990). *Identity, consciousness and value*. Oxford University Press. https://doi.org/10.1093/oso/9780195054019.001.0001

Vogt, B. A. (2016). Midcingulate cortex: Structure, connections, homologies, functions and diseases. *Journal of Chemical Neuroanatomy*, *74*, 28–46. https://doi.org/10.1016/j.jchemneu.2016.01.010

Westlin, C., Theriault, J. E., Katsumi, Y., Nieto-Castanon, A., Kucyi, A., Ruf, S. F., Brown, S. M., Pavel, M., Erdogmus, D., Brooks, D. H., Quigley, K. S., Whitfield-Gabrieli, S., & Barrett, L. F. (2023). Improving the study of brain-behavior relationships by revisiting basic assumptions. *Trends in Cognitive Sciences*, 27(3), 246–257. https://doi.org/10.1016/j.tics.2022.12.015

Xu, R., Lv, P., Xu, F., & Shi, Y. (2021). A survey of approaches for implementing optical neural networks. *Optics & Laser Technology*, *136*, 106787. https://doi.org/10.1016/j.optlastec.2020.106787

Yu, G. I., Jeong, J. S., Kim, G. W., Kim, S., & Chun, B. G. (2022). ORCA: A distributed serving system for transformer-based generative models. *Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation*, 521–538.

Zidan, M. A., Strachan, J. P., & Lu, W. D. (2018). The future of electronics based on memristive systems. *Nature Electronics*, *1*(1), 22–29. https://doi.org/10.1038/s41928-017-0006-8

Zilles, K., & Amunts, K. (2010). Centenary of Brodmann's map — Conception and fate. *Nature Reviews Neuroscience*, *11*(2), 139–145. https://doi.org/10.1038/nrn2776